Nearline Web Archiving

Zhiwu Xie[1], Krati Nayyar[2], and Edward A. Fox[3]
[1]University Libraries, [2]Department of Electrical & Computer Engineering, [3]Department of Computer Science;
Virginia Tech, Blacksburg, VA 24061

Based on the acquisition method, web archiving may be categorized into client-side, transactional, and server-side archiving [1]. Transactional web archiving happens at the gateway of the origin server. As the name suggests, it archives the HTTP responses to user requests, typically in real time, when the HTTP transactions occur [2, 3]. Despite its distinctive temporal coverage, transactional web archiving suffers an inherent technical disadvantage. Similar to the server-side archiving, it requires the cooperation of the website owner and/or operator to install server-side add-ons. Much like the client-side archiving, it also hinges on the HTTP protocol and its “inability to provide bulk copy of server’s content” [1]. Archiving web documents one-by-one inevitably injects extra workload directly onto the origin server, making it rather difficult to seek the owner's cooperation.

In this paper, we propose a modified approach to real-time transactional web archiving. It leverages the web caching infrastructure that is already prevalent on web servers. Instead of archiving web content at HTTP transaction time, in our approach the archiving happens when the cached copy expires and is about to be expunged. Before the deletion, all expired cache copies are combined and then sent to the web archive in small batches. Since the cache is purged at much lower frequency than HTTP transactions, the archival workload is also much lower than that for transactional archiving. To further decrease the processing load at the origin server, archival copy deduplication is carried out at the archive instead of at the origin server. It is crucial to note that the cache purging process is separate from those that serve the HTTP requests. It can be, and usually is set to lower priority. The archiving therefore occurs only when the server is not busy fulfilling its more mission critical tasks; this is much less disruptive to the origin server. This approach, however, does not guarantee that the freshest copy is archived, although the cache purging policy may be adjusted to attempt to bound the freshness of the archive.

Borrowing a term from the storage method, we call this approach “nearline” web archiving. It does not happen online at transaction time. Neither does it happen fully offline as in the server-side archiving, which is usually in large batches with no regard for user requests. Instead it observes the transaction, and then acts at a different, much slower pace. This approach retains much of the quality of transactional archiving, but has better performance.

We provide a prototypical implementation of the nearline archiving based on the Apache HTTP server. We modified htcacheclean.c, the C code that will be evoked to clean the Apache disk cache if configured. In our implementation, the files to be deleted are written into a WARC file and uploaded to an external web archive. The source code may be found at http://github.com/VTUL

Implementation Steps:

Apache HTTP project provides the functionality of implementing a disk based storage manager. This can be done by enabling the mod_cache_disk module in cache_disk.conf configuration file. It also provides a tool called htcacheclean to maintain the size of the disk cache within limits or to delete the cache all together. The tool can be run in the daemon mode after regular intervals or manual for running it only once.

The idea proposed in this paper executes when the htcacheclean command is run and stores all the cache being deleted in the WARC files before deleting the cache files from the server disk cache.
Wget library , used for retrieving content from web servers and written in C language, has the functionality of retrieving content of a specific URL and making the WARC files depending on the command line arguments given. Wget code is taken as the reference to develop the code for making warc files from the content of the disk server cache files.
As the cache files only consist of HTTP response, the type of WARC record which is generated by the code is “response” till this point.

References

[1] Masanès, Julien. Web archiving. Berlin: Springer, 2006.
[2] Van de Sompel, H. "SiteStory transactional web archive software released." D-Lib Magazine 18.9/10 (2012). http://www.dlib.org/dlib/september12/09inbrief.html
[3] Brunelle, J.F., Nelson, M.L., Balakireva, L., Sanderson, R., Van de Sompel, H. “Evaluating the SiteStory transactional web archive with the ApacheBench tool”. In: 17th Annual Conference on the Theory and Practice of Digital Libraries, pp. 204–215 (2012).